

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A system for creating query refinement suggestions, comprising:

a database configured to store content of a plurality of documents;

a log configured to store received search queries;

a server configured to receive a current search query from a user and identify at least one search document stored in the database based on the current search query;

a matcher ~~matching~~ configured to match the at least one search document ~~retrieved responsive to a query~~ to one or more of the stored search queries from the log, wherein the current search query differs from the matched one or more stored search queries; and

a scorer ~~scoring~~ configured to score the stored query matched one or more search queries as a potential query refinement suggestion to be presented to the user.

2. (currently amended) A system according to claim 1, further comprising:

a document matcher ~~matching~~ configured to match the at least one search document to one or more stored documents associated with the matched one or more stored search queries.

3. (currently amended) A system according to claim 1, further comprising:

a weight associated with at least one ~~[[such]]~~ of the stored ~~query~~ search queries;
and

a clusterer ~~forming~~ configured to form at least one cluster based on the at least one stored search query and the associated weight.

4. (original) A system according to claim 3, further comprising:
a term vector used in cluster formation computed from terms extracted from the stored query and based on the weight for the stored query.

5. (currently amended) A system according to claim 4, further comprising:
a distance for the term vector; and
the cluster ~~forming~~ configured to form the at least one cluster relative to the distance.

6. (currently amended) A system according to claim 3, further comprising:
a ranker ~~ranking~~ configured to rank the at least one cluster relative to the at least one other cluster by evaluating a relevance score associated with each search document corresponding to each matched stored document.

7. (currently amended) A system according to claim 6, further comprising:
a selector ~~selecting~~ configured to select at least one ranked cluster as a potential refinement cluster.

8. (original) A system according to claim 3, further comprising:
a centroid computed as a weighted center of the at least one cluster; and
a score computed for each stored query relative to the centroid.
9. (currently amended) A system according to claim 8, further comprising:
a selector ~~naming~~ configured to name the at least one cluster for at least one
scored stored query.
10. (currently amended) A system according to claim 1, further comprising:
a threshold applied to the ~~stored~~ scored query one or more search queries.
11. (currently amended) A system according to claim 1, further comprising:
a precomputation engine ~~associating~~ configured to associate one or more stored
documents to the one or more stored ~~query~~ search queries based on at least one of a
chosen search document, a set of search documents, regenerated previous search
documents, ~~[[and]]~~ or inverted cached document and query pairings.
12. (currently amended) A method for creating query refinement suggestions,
comprising:
storing documents in a database,
storing a plurality of queries received via a network;
receiving a search query from a user;

identifying at least one search document from the database based on the search query;
matching the at least one search document ~~retrieved responsive to a query~~ to one or more of the stored queries, wherein the search query differs from the matched one or more stored queries; and
scoring the matched one or more stored ~~query~~ queries as a potential query refinement suggestion to be presented to the user.

13. (currently amended) A method according to claim 12, further comprising:
matching the at least one search document to one or more stored documents associated with the matched one or more stored queries.

14. (currently amended) A method according to claim 12, further comprising:
associating a weight with at least one ~~[[such]]~~ stored query; and
forming at least one cluster based on the at least one stored query and the associated weight.

15. (currently amended) A method according to claim 14, further comprising:
computing a term vector used in cluster formation from terms extracted from the at least one stored query and based on the associated weight ~~for the stored query.~~

16. (original) A method according to claim 15, further comprising:
determining a distance for the term vector; and

forming the at least one cluster relative to the distance.

17. (original) A method according to claim 14, further comprising:
ranking the at least one cluster relative to the at least one other cluster by
evaluating a relevance score associated with each search document corresponding to each
matched stored document.

18. (original) A method according to claim 17, further comprising:
selecting at least one ranked cluster as a potential refinement cluster.

19. (original) A method according to claim 14, further comprising:
computing a centroid as a weighted center of the at least one cluster; and
computing a score for each stored query relative to the centroid.

20. (original) A method according to claim 19, further comprising:
naming the at least one cluster for at least one scored stored query.

21. (currently amended) A method according to claim 12, further comprising:
applying a threshold to the one or more stored scored ~~query~~ queries.

22. (currently amended) A method according to claim 12, further comprising:
associating one or more stored documents to the one or more stored ~~query~~ queries
based on at least one of a chosen search document, a set of search documents,

regenerated previous search documents, and inverted cached document and query pairings.

23. (original) A computer-readable storage medium holding code for performing the method according to claim 12.

24. (currently amended) An apparatus for creating query refinement suggestions, comprising:

means for storing a plurality of documents;

means for storing a plurality of queries;

means for matching, to one or more stored queries, at least one search document retrieved responsive to a search query to one or more stored queries received from a user, wherein the received search query and the matched one or more stored queries differ; and

means for scoring the matched one or more stored query queries as a potential query refinement suggestion to be presented to the other user.

25. (currently amended) A system for providing search query refinements, comprising:

an associator associating configured to associate a previously received, stored query and a stored document as a logical pairing and assigning assign a weight to the logical pairing;

a searcher issuing the configured to receive a search query from a user and producing produce a set of search documents based on the received search query;

a matcher ~~matching~~ configured to match at least one of the set of search ~~document~~ documents to at least one stored document and ~~retrieving~~ retrieve the stored query and the assigned weight associated with the matching at least one stored document, wherein the received search query and the stored query differ;

a clusterer ~~forming~~ configured to form at least one cluster based on the stored query and the assigned weight associated with the matching at least one stored document; and

a scorer ~~scoring~~ configured to score the stored query associated with the matching at least one stored document for the at least one cluster relative to at least one other cluster and ~~suggesting~~ suggest at least one [[such]] scored search query as a set of query refinements to be presented to the user.

26. (currently amended) A system according to claim 25, further comprising:

a selector ~~selecting~~ configured to select one [[such]] search document chosen from among the set of search documents responsive to the search query issuance as the at least one [[such]] search document.

27. (currently amended) A system according to claim 25, further comprising:

a selector ~~selecting~~ configured to select the set of search documents as the at least one [[such]] search document.

28. (currently amended) A system according to claim 25, further comprising:

a query log ~~tracking~~ configured to track previous search queries; and

a regenerator ~~regenerating~~ configured to regenerate a set of previous search documents produced by the previous search queries as the at least one [[such]] search document.

29. (currently amended) A system according to claim 25, further comprising:
a cache ~~associating~~ configured to associate at least one cached document and one or more cached queries as a cached pairing; and
an inverter ~~inverting~~ configured to invert each cached pairing to associate at least one cached query and one or more cached documents as the at least one [[such]] search document.

30. (original) A system according to claim 25, wherein relevancy to the stored query is estimated for the stored document as the weight assigned to the pairing.

31. (currently amended) A system according to claim 30, wherein each [[such]] assigned weight for a plurality of pairings corresponding to the stored query and the stored document is summed.

32. (currently amended) A system according to claim 25, wherein each stored query comprises one or more terms, further comprising:

a term vector comprising the terms in the stored query associated with the matching at least one stored document;

a distance determined for the term vector; and

the clusterer ~~forming~~ to form the at least one cluster relative to the distance.

33. (currently amended) A system according to claim 32, further comprising:
a normalizer ~~normalizing~~ configured to normalize the term vector.

34. (currently amended) A system according to claim 32, further comprising:
an evaluator ~~computing~~ configured to commute a length of the term vector in
multi-dimensional space with each dimension equaling a sum of the weights of the term
in a set of associated stored queries.

35. (currently amended) A system according to claim 32, further comprising:
a relevance score assigned to the at least one search document; and
a ranker ~~ranking~~ configured to rank the at least one cluster relative to the at least
one other cluster by the relevance score associated with the matching at least one search
document and a number of the matching at least one search document.

36. (currently amended) A system according to claim 35, further comprising:
a selector ~~selecting~~ configured to select one of more of the ranked at least one
cluster as potential refinement clusters based on the rankings.

37. (currently amended) A system according to claim 36, further comprising:
a centroid computed as a weighted center for each ~~[[such]]~~ potential refinement
cluster; and

the scorer ~~scoring~~ to score the stored query associated with the matching at least one stored document for the potential refinement cluster relative to the centroid.

38. (currently amended) A system according to claim 37, further comprising:
an evaluator ~~computing~~ configured to compute the centroid as a normalized sum of a product of the term vector for each stored query and the relevance score associated with the matching at least one search document.

39. (currently amended) A system according to claim 38, further comprising:
a length of a distance vector determined from the term vector and the centroid;
and

the scorer ~~computing~~ configured to compute the score for the scored query as a product of a number of stored documents with which the stored query is associated and the distance vector length.

40. (currently amended) A system according to claim 25, further comprising:
a selector ~~selecting~~ configured to select the stored query associated with the matching at least one stored document relative to a threshold.

41. (currently amended) A system according to claim 25, further comprising:
a sorter ~~sorting~~ configured to sort the set of query refinements.

42. (currently amended) A system according to claim 25, further comprising:

a presenter ~~presenting~~ configured to present the set of query refinements.

43. (original) A system according to claim 25, further comprising:

a set of supplemental query refinements negating each term in the set of query refinements not present in the search query and using the negated terms in combination with the search query as at least one supplemental query refinement.

44. (currently amended) A system according to claim 25, further comprising:

an association database ~~maintaining~~ to maintain the pairings.

45. (currently amended) A system according to claim 25, wherein at least one of each ~~[[such]]~~ stored document and each ~~[[such]]~~ search document is specified as at least one of a Uniform Resource Locator (URL), hyperlink, anchor, ~~[[and]]~~ or document excerpt.

46. (currently amended) A method for providing search query refinements, comprising:

associating a stored query and a stored document as a logical pairing and
assigning a weight to the logical pairing;

~~issuing the~~ receiving a search query from a user and producing a set of search documents based on the received search query;

matching at least one of the set of search ~~document~~ documents to at least one stored document and retrieving the stored query and the assigned weight associated with

the matching at least one stored document, wherein the received search query and the stored query differ;

forming at least one cluster based on the stored query and the assigned weight associated with the matching at least one stored document; and

scoring the stored query associated with the matching at least one stored document for the at least one cluster relative to at least one other cluster and suggesting at least one [[such]] scored search query as a set of query refinements to the user.

47. (currently amended) A method according to claim 46, further comprising:
selecting one [[such]] search document chosen from among the set of search documents responsive to the search query issuance as the at least one [[such]] search document.

48. (currently amended) A method according to claim 46, further comprising:
selecting the set of search documents as the at least one [[such]] search document.

49. (currently amended) A method according to claim 46, further comprising:
tracking previous search queries; and
regenerating a set of previous search documents produced by the previous search queries as the at least one [[such]] search document.

50. (currently amended) A method according to claim 46, further comprising:

associating at least one cached document and one or more cached queries as a cached pairing; and

inverting each cached pairing to associate at least one cached query and one or more cached documents as the at least one [[such]] search document.

51. (currently amended) A method according to claim 46, further comprising:
for each [[such]] pairing, estimating relevancy to the stored query for the stored document as the weight assigned to the pairing.

52. (currently amended) A method according to claim 51, further comprising:
summing each [[such]] assigned weight for a plurality of pairings corresponding to the stored query and the stored document.

53. (original) A method according to claim 46, wherein each stored query comprises one or more terms, further comprising:
computing a term vector comprising the terms in the stored query associated with the matching at least one stored document;
determining a distance determined for the term vector; and
forming the at least one cluster relative to the distance.

54. (original) A method according to claim 53, further comprising:
normalizing the term vector.

55. (original) A method according to claim 53, further comprising:

computing a length of the term vector in multi-dimensional space with each

dimension equaling a sum of the weights of the term in a set of associated stored queries.

56. (original) A method according to claim 53, further comprising:

assigning a relevance score to the at least one search document; and

ranking the at least one cluster relative to the at least one other cluster by the

relevance score associated with the matching at least one search document and a number

of the matching at least one search document.

57. (original) A method according to claim 56, further comprising:

selecting one of more of the ranked at least one cluster as potential refinement

clusters based on the rankings.

58. (currently amended) A method according to claim 57, further comprising:

computing a centroid as a weighted center for each [[such]] potential refinement

cluster; and

scoring the stored query associated with the matching at least one stored

document for the potential refinement cluster relative to the centroid.

59. (original) A method according to claim 58, further comprising:

computing the centroid as a normalized sum of a product of the term vector for

each stored query and the relevance score associated with the matching at least one

search document.

60. (original) A method according to claim 59, further comprising:
determining a length of a distance vector from the term vector and the centroid;
and
computing the score for the scored query as a product of a number of stored
documents with which the stored query is associated and the distance vector length.

61. (original) A method according to claim 46, further comprising:
selecting the stored query associated with the matching at least one stored
document relative to a threshold.

62. (original) A method according to claim 46, further comprising:
sorting the set of query refinements.

63. (original) A method according to claim 46, further comprising:
presenting the set of query refinements.

64. (original) A method according to claim 46, further comprising:
negating each term in the set of query refinements not present in the search query
and using the negated terms in combination with the search query as at least one
supplemental query refinement.

65. (original) A method according to claim 46, further comprising:
maintaining the pairings in a database.
66. (currently amended) A method according to claim 46, further comprising:
specifying at least one of each ~~[[such]]~~ stored document and each ~~[[such]]~~ search document as at least one of a Uniform Resource Locator (URL), hyperlink, anchor, ~~[[and]]~~ or document excerpt.
67. (original) A computer-readable storage medium holding code for performing the method according to claim 46.
68. (currently amended) An apparatus for providing search query refinements, comprising:
means for associating a stored query and a stored document as a logical pairing and means for assigning a weight to the logical pairing;
means for ~~issuing the~~ receiving a search query from a user and means for producing a set of search documents based on the received search query;
means for matching at least one of the set of search ~~document~~ documents to ~~at least one~~ the stored document and means for retrieving the stored query and the assigned weight associated with the matching ~~at least one~~ stored document, wherein the stored query and the received search query differ;
means for forming at least one cluster based on the stored query and the assigned weight associated with the matching at least one stored document; and

means for scoring the stored query associated with the matching ~~at least one~~ stored document for the at least one cluster relative to at least one other cluster and means for suggesting at least one ~~[[such]]~~ scored search query as a set of query refinements to be presented to the user.

69. (currently amended) A system for integrating query refinement candidates, comprising:

a matcher ~~matching~~ configured to match, to one or more stored documents associated with a previously received, a stored query having an associated weight, at least one search document retrieved responsive to a query from a user ~~to one or more stored documents associated with a stored query and weight~~ and ~~matching~~ match at least one further search document retrieved responsive to a candidate query to the one or more stored documents, wherein the query from the user and the stored query differ;

a ~~cluster forming~~ clusterer configured to form at least one cluster based on the stored query and weight associated with each stored document matched responsive to the query and ~~forming~~ form at least one further cluster based on the stored query and weight associated with each stored document matched responsive to the candidate query;

a combiner ~~combining~~ configured to combine the at least one cluster and the at least one further cluster; and

a scorer ~~scoring~~ configured to score the stored query for the combined cluster relative to at least one other cluster as a potential query refinement suggestion to be presented to the user.

70. (currently amended) A system according to claim 69, further comprising:
a set of candidate query refinements comprising at least one ~~[[such]]~~ candidate query.

71. (currently amended) A system according to claim 70, further comprising:
an evaluator ~~assigning~~ configured to assign at least one ~~[[such]]~~ candidate query to the at least one cluster.

72. (currently amended) A system according to claim 71, further comprising:
a builder ~~creating~~ configured to create an orthogonal set of candidate query refinements comprising at least one ~~[[such]]~~ unassigned query candidate.

73. (currently amended) A method for integrating query refinement candidates, comprising:
matching, to one or more stored documents associated with a previously received, stored query having an associated weight, at least one search document retrieved responsive to a query to one or more stored documents associated with a stored query and weight received from a user;

forming at least one cluster based on the stored query and weight associated with each stored document matched responsive to the query;

matching at least one further search document retrieved responsive to a candidate query to the one or more stored documents;

forming at least one further cluster based on the stored query and weight associated with each stored document matched responsive to the candidate query; combining the at least one cluster and the at least one further cluster; and scoring the stored query for the combined cluster relative to at least one other cluster as a potential query refinement suggestion.

74. (currently amended) A method according to claim 73, further comprising: assembling a set of candidate query refinements comprising at least one [[such]] candidate query.

75. (currently amended) A method according to claim 74, further comprising: assigning at least one [[such]] candidate query to the at least one cluster.

76. (currently amended) A method according to claim 75, further comprising: creating an orthogonal set of candidate query refinements comprising at least one [[such]] unassigned query candidate.

77. (original) A computer-readable storage medium holding code for performing the method according to claim 73.

78. (currently amended) An apparatus for integrating query refinement candidates, comprising:

means for matching, to one or more stored documents associated with a previously received, stored query having an associated weight, at least one search document retrieved responsive to a query ~~to one or more stored documents associated with a stored query and weight~~;

means for forming at least one cluster based on the stored query and weight associated with each stored document matched responsive to the query;

means for matching at least one further search document retrieved responsive to a candidate query to the one or more stored documents;

means for forming at least one further cluster based on the stored query and weight associated with each stored document matched responsive to the candidate query;

means for combining the at least one cluster and the at least one further cluster;
and

means for scoring the stored query for the combined cluster relative to at least one other cluster as a potential query refinement suggestion.